**Batch: IAI-2 Experiment Number: 8**

**Roll Number: 16010422234 Name: Chandana Ramesh Galgali**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Aim of the Experiment:** To implement Decision Tree Algorithm (ID3 using library functions)

**Program/Steps:**

Set up and train a decision tree classifier on the Titanic dataset and see how well the classifier performs on a validation set (80-20 train-test dataset). Find out accuracy and confusion matrix and plot created decision tree with following variations

1. Target Variable: Survived , remaining all input features

2. Target Variable: Survived , selecting subset of features as input

3. Target Variable: Survived , using transformed input feature ( e.g. create new feature family =

sibsp + parch, weighted\_class = pclass\*2 if pclass =1 ; pclass\*3 if pclass =2 ; pclass\*4 if pclass

=3 etc)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Program:**

**import pandas as pd**

**from sklearn.model\_selection import train\_test\_split**

**from sklearn.tree import DecisionTreeClassifier**

**from sklearn.metrics import accuracy\_score, confusion\_matrix**

**from sklearn.tree import plot\_tree**

**import matplotlib.pyplot as plt**

**# Load the dataset**

**url = "https://web.stanford.edu/class/archive/cs/cs109/cs109.1166/stuff/titanic.csv"**

**data = pd.read\_csv(url)**

**# Preprocessing**

**data['Age'].fillna(data['Age'].median(), inplace=True)**

**# Convert 'Sex' to a numeric variable**

**data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})**

**# Variation-1: Using All Input Features**

**# Define features and target**

**features = data.drop(['Survived', 'Name'], axis=1) # Exclude 'Name' for being non-predictive**

**target = data['Survived']**

**# Train-test split**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(features, target, test\_size=0.2, random\_state=42)**

**# Decision Tree Classifier**

**classifier = DecisionTreeClassifier()**

**classifier.fit(X\_train, y\_train)**

**predictions = classifier.predict(X\_test)**

**print("Using All Input Features:")**

**# Metrics**

**print("Accuracy:", accuracy\_score(y\_test, predictions))**

**print("Confusion Matrix:\n", confusion\_matrix(y\_test, predictions))**

**# Plot**

**plt.figure(figsize=(20,10))**

**plot\_tree(classifier, filled=True, feature\_names=features.columns, class\_names=['Died', 'Survived'], rounded=True, fontsize=12)**

**plt.show()**

**# Variation-2: Using a Subset of Features**

**features\_subset = data[['Age', 'Fare', 'Sex']] # Selected subset of features**

**# Train-test split**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(features\_subset, target, test\_size=0.2, random\_state=42)**

**# Decision Tree Classifier**

**classifier\_subset = DecisionTreeClassifier()**

**classifier\_subset.fit(X\_train, y\_train)**

**predictions\_subset = classifier\_subset.predict(X\_test)**

**print("\nUsing a Subset of Features:")**

**# Metrics**

**print("Accuracy:", accuracy\_score(y\_test, predictions\_subset))**

**print("Confusion Matrix:\n", confusion\_matrix(y\_test, predictions\_subset))**

**# Plot**

**plt.figure(figsize=(20,10))**

**plot\_tree(classifier\_subset, filled=True, feature\_names=features\_subset.columns, class\_names=['Died', 'Survived'], rounded=True, fontsize=12)**

**plt.show()**

**# Variation-3: Using Transformed Input Features**

**# Feature engineering**

**data['Family'] = data['Siblings/Spouses Aboard'] + data['Parents/Children Aboard']**

**data['Weighted\_Class'] = data['Pclass'].apply(lambda x: x\*2 if x == 1 else (x\*3 if x == 2 else x\*4))**

**# Define features with transformations**

**features\_transformed = data[['Age', 'Fare', 'Sex', 'Family', 'Weighted\_Class']]**

**# Train-test split**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(features\_transformed, target, test\_size=0.2, random\_state=42)**

**# Decision Tree Classifier**

**classifier\_transformed = DecisionTreeClassifier()**

**classifier\_transformed.fit(X\_train, y\_train)**

**predictions\_transformed = classifier\_transformed.predict(X\_test)**

**print("\nUsing Transformed Input Features:")**

**# Metrics**

**print("Accuracy:", accuracy\_score(y\_test, predictions\_transformed))**

**print("Confusion Matrix:\n", confusion\_matrix(y\_test, predictions\_transformed))**

**# Plot**

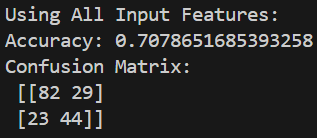
**plt.figure(figsize=(20,10))**

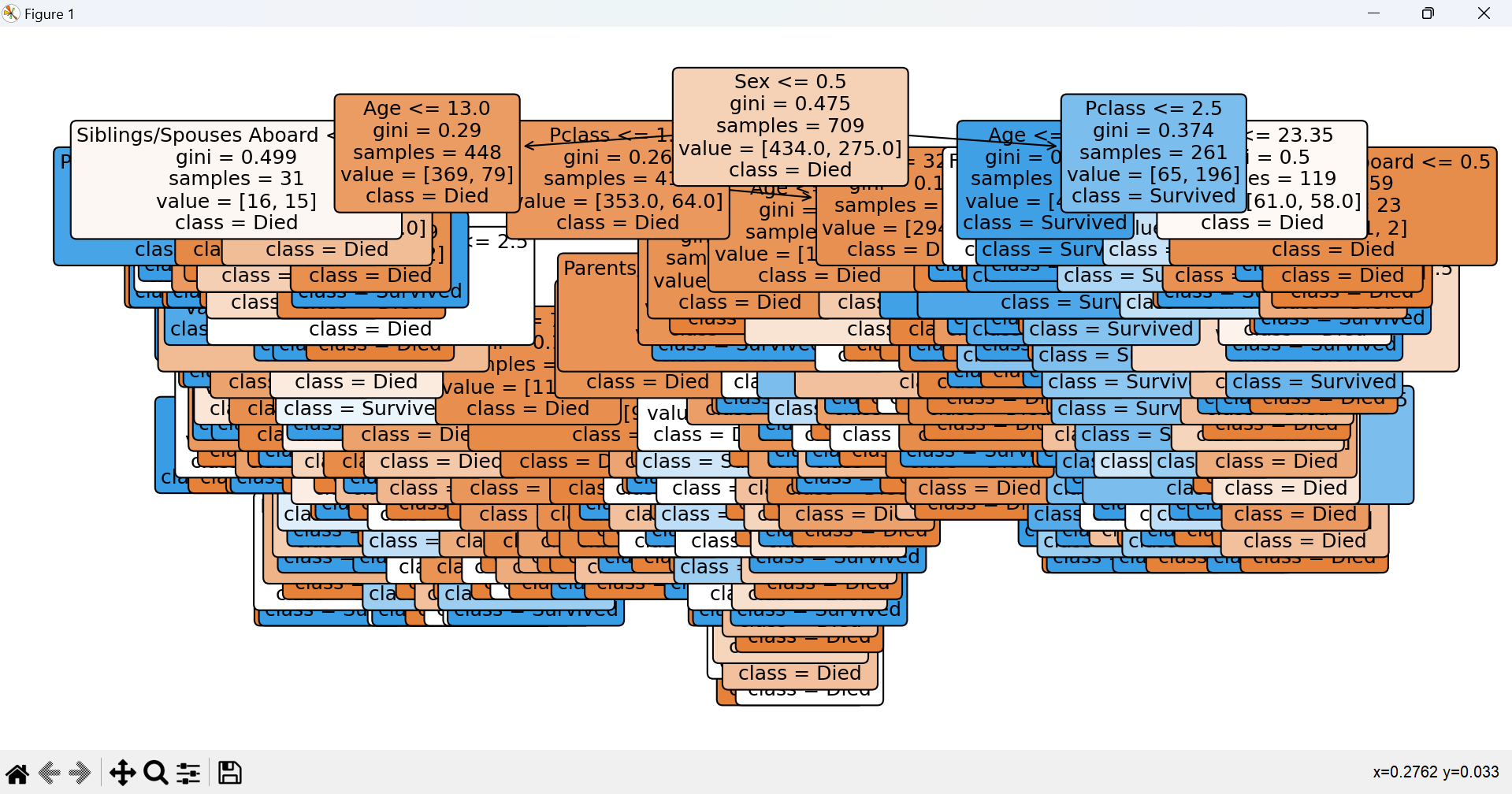
**plot\_tree(classifier\_transformed, filled=True, feature\_names=features\_transformed.columns, class\_names=['Died', 'Survived'], rounded=True, fontsize=12)**

**plt.show()**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

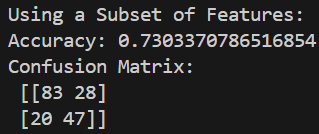
**Output/Result:** Variation-1

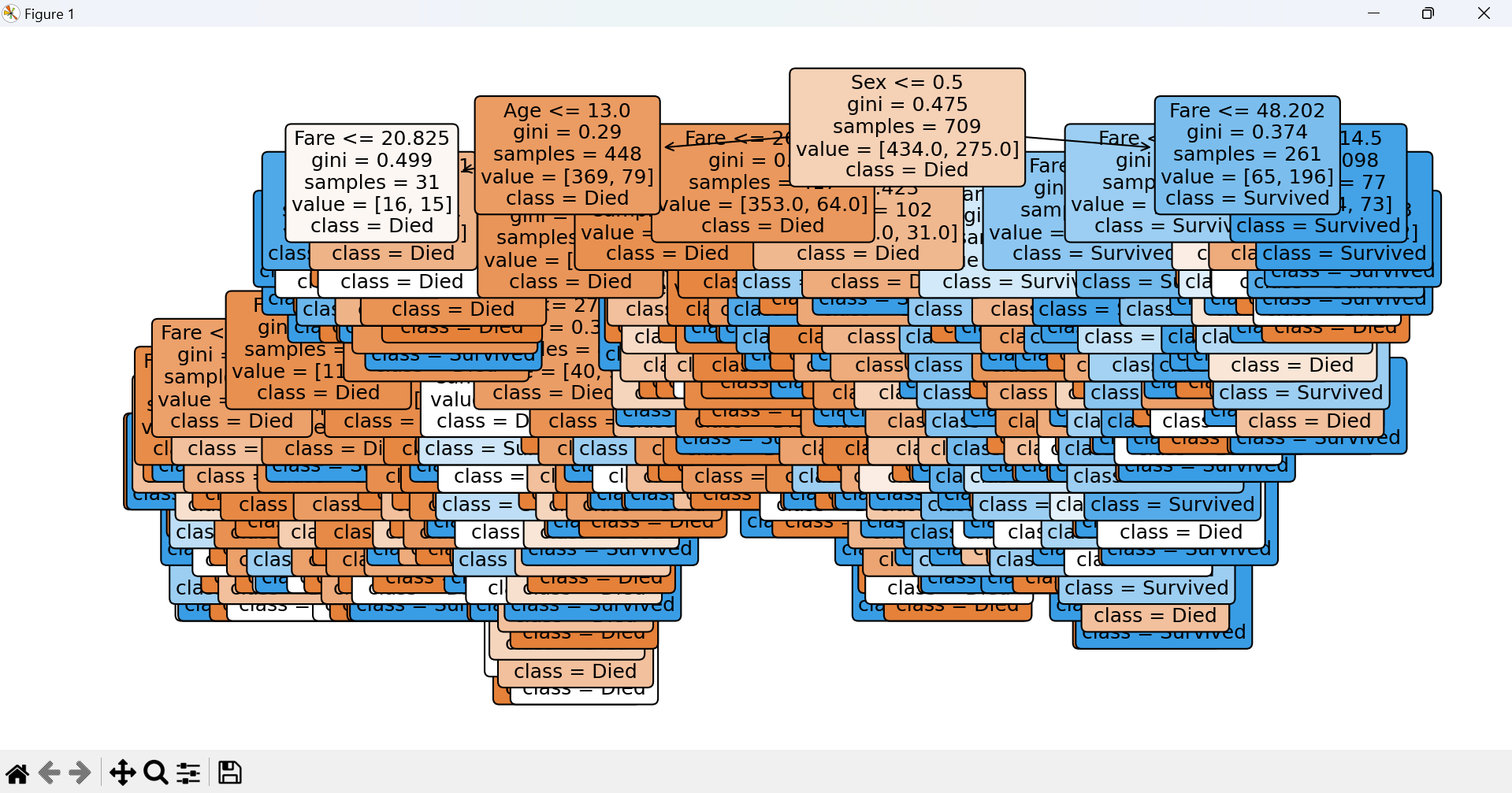
****

****

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

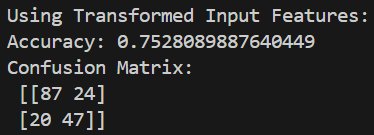
Variation-2

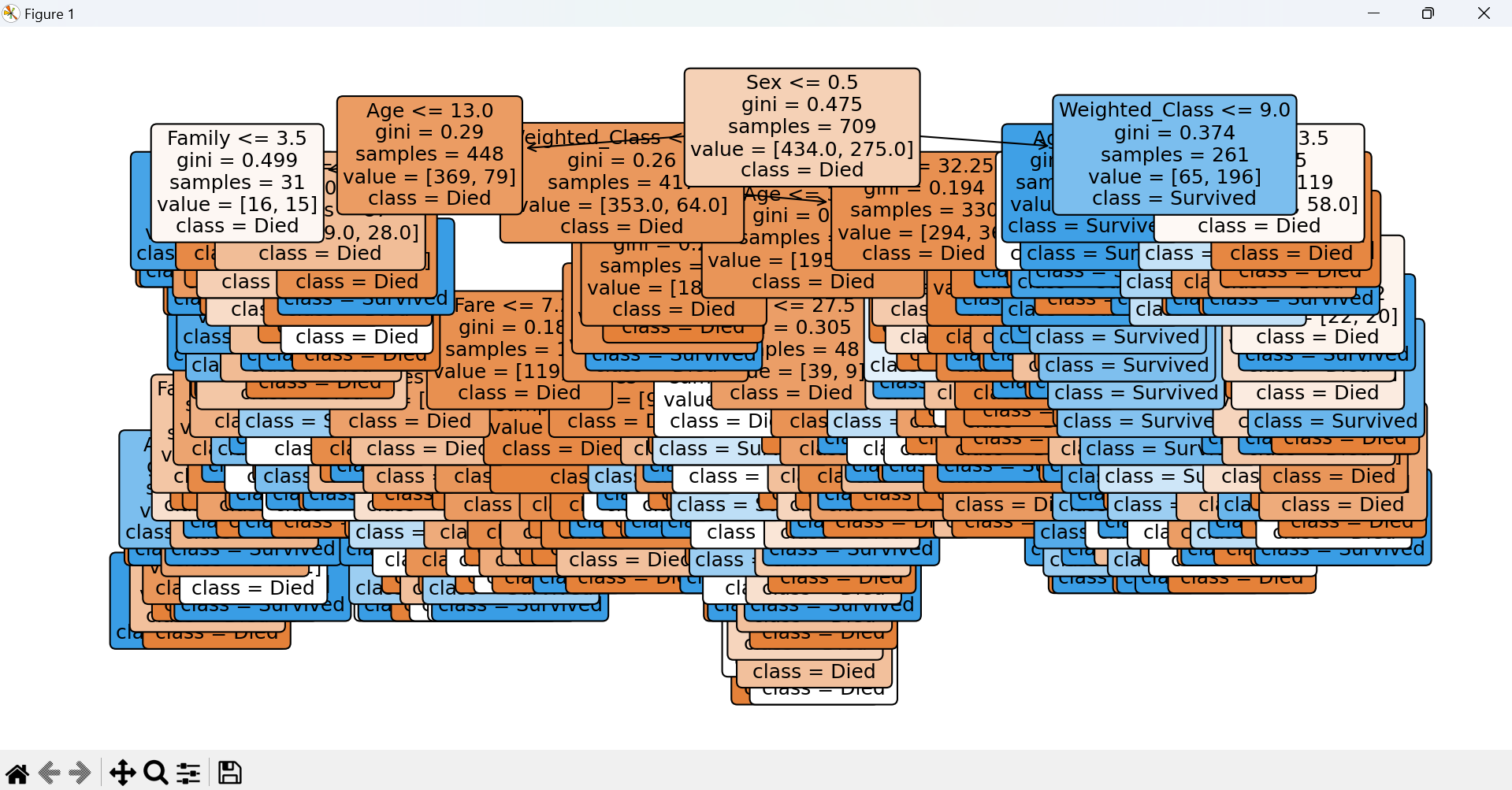
****

****

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Variation-3

****

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Outcomes: Understand fundamentals of learning in AI.**

**Conclusion (Based on the Results and outcomes achieved):**

Using the decision tree model on the Titanic dataset allows for straightforward interpretation of survival factors through a visual representation. The accuracy metric and confusion matrix provide quantitative measures of the model's performance, highlighting its strengths and areas for improvement. Feature engineering played a crucial role in enhancing the model's predictive power by introducing new, meaningful features based on existing data. This method offers a robust approach to understanding complex patterns in the data, making it a valuable tool for both predictive modeling and exploratory data analysis.

**References:**

1. Stuart Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, Second Edition, Pearson Publication
2. Jason Brownlee , Master Machine Learning Algorithms, eBook, 2017 ,Edition v1.12.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_